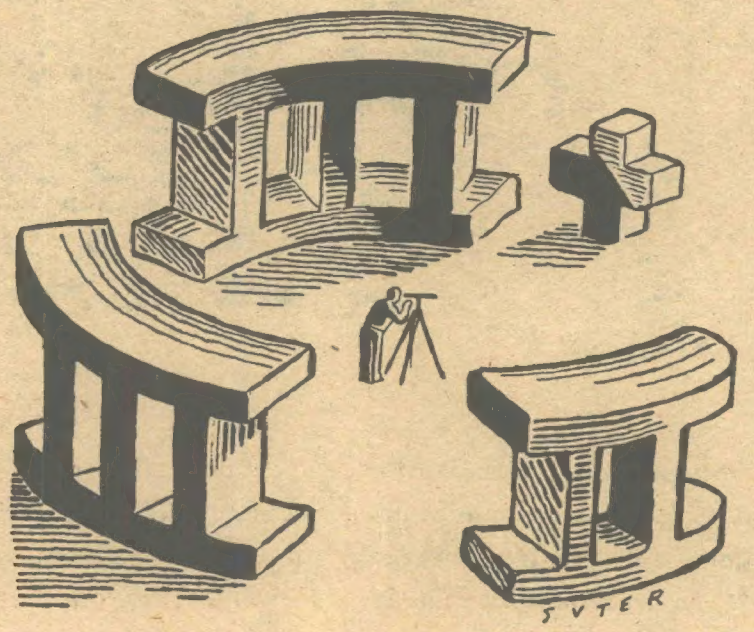
จะเลือกใช้ dBASE เวอร์ชันไหนดี



โบราณว่า "เก่า ๆ มันเป็นสนิม..... ใหม่กว่าหน้าตาจุ่มจิ่ม" ดีเบสก็เข้าทำนองนี้ แล้วเราจะเลือกใช้ดีเบสเวอร์ชันไหนดี จึงจะเกิดประโยชน์สูงสุดแก่งาน

อารีรัตน์ ศิริคุณ

มาจนบัดนี้ บริษัทแอชตัน–เทต ได้พัฒนาโปรแกรมในตระกูลดีเบสออกมาถึง 3 เวอร์ชันแล้ว ได้แก่ ดีเบสทู ดีเบสทรี และดีเบสทรีพลัส การที่จะตัดสินว่าเวอร์ชันไหนใช้งานได้ดีที่สุดนั้น จะต้องนำส่วนอื่น ๆ มาพิจารณาประกอบด้วย เช่น ฮาร์ดแวร์ ระบบการทำงานของเครื่อง (Operating System), ขนาดและความซับซ้อนของฐานข้อมูลที่มีอยู่แล้ว และที่คาดว่าจะต้องนำไปขยับขยายใช้ในอนาคต

สำหรับผู้ที่ใช้ระบบ CP/M และยังไม่คิดจะเปลี่ยนฮาร์ดแวร์มาเป็น 16 บิทแล้ว ดีเบสทูก็เป็นทางเลือกเพียงทางเดียว แต่สำหรับผู้ที่มีเครื่องไอบีเอ็ม หรือเครื่องคอมแพต และยังอยู่ในระหว่างการตัดสินใจว่าจะใช้ดีเบสเวอร์ชันไหนดี ก็คงจะลงเอยกับดีเบสทรีพลัส เพราะจะว่าไปแล้ว ดีเบสทรีพลัสก็ถูกพัฒนาขึ้นมาเพื่อใช้แทนดีเบสทรี เหมือนกับคราวที่แอชตัน–เทตออกดีเบสทู 2.4 มาแทนดีเบสทู 2.3 ส่วนผู้ที่ต้องการจะสร้างระบบความปลอดภัยของข้อมูล และต้องการสร้างระบบฐานข้อมูลที่ผู้ใช้หลายคนสามารถเรียกใช้ข้อมูลร่วมกันได้ (Multi-user) ก็คงจะต้องใช้บริการของดีเบสทรีพลัส เพราะเป็นเวอร์ชันเดียวที่สามารถใช้กับระบบเครือข่ายไมโคร–คอมพิวเตอร์ได้ (Local Area Network)

สำหรับผู้ที่ใช้เครื่องไอบีเอ็ม หรือเครื่องคอมแพตอยู่แล้ว หากคิดจะเปลี่ยนจากดีเบสทูมาใช้ดีเบสทรี หรือเปลี่ยนจากดีเบสทรีมาเป็นดีเบสทรีพลัสเป็นเรื่องไม่ยากเย็นอะไรเลย เพียงแต่ว่าผู้ใช้งานบางคนรู้สึกขยาดกับการที่จะต้องมาเรียนรู้คำสั่งใหม่ ๆ เพิ่มเติม ซึ่งสำหรับบางคนกลับเป็นเรื่องสนุกดีเสียอีกที่ได้ปรับตัวให้ทันสมัยกับเวอร์ชันใหม่ ๆ อย่างไรก็ดี การเปลี่ยนจากดีเบสทูมาใช้ดีเบสทรีหรือดีเบสทรีพลัสนั้น ควรจะพิจารณาให้ดีเสียก่อนว่า การแปลงไฟล์หรือโปรแกรมที่มีอยู่เพื่อมาใช้กับดีเบสเวอร์ชันที่ใหม่กว่านั้นจะคุ้มกับเงินและเวลาที่จะลงไปหรือไม่

จากดีเบสทูมาเป็นดีเบสทรี

การเปลี่ยนจากดีเบสทูมาใช้ดีเบสทรีนั้นเรียกได้ว่าเป็นการเปลี่ยนแบบก้าวกระโดดทีเดียว ในขณะที่การเปลี่ยนจากดีเบสทรีมาเป็นดีเบสทรีพลัสนั้น ถ้าไม่รวมไปถึงการใช้งานในระบบมัลติยูสเซอร์ในดีเบสทรีพลัสแล้ว ก็นับว่าเป็นเพียงการปรับปรุงรูปโฉมจากเดิมเท่านั้นเอง

การเปลี่ยนจากดีเบสทูมาใช้ดีเบสทรีหรือดีเบสทรีพลัสนั้น ผู้ใช้จะพบว่าคำสั่งและโครงสร้างของภาษาที่ใช้เปลี่ยนไป

จากเดิมมาก แต่หลังจากที่ใช้งานจนชินแล้วจะเห็นว่าเป็นการเปลี่ยนแปลงที่ผู้ใช้ได้รับประโยชน์มากขึ้น อย่างไรก็ตาม การเรียนรู้คำสั่งหรือฟังก์ชันใหม่ๆ ในดีเบสทรีหรือดีเบสทรีพลัสนั้นต้องใช้เวลาพอสมควร และเมื่อต้องการนำโปรแกรมเฉพาะงานที่ใช้ในดีเบสทูมาใช้กับดีเบสทรีหรือดีเบสทรีพลัสนั้น ก็ต้องมีการแก้ไขดัดแปลงโปรแกรมนั้นๆเสียก่อน ซึ่งผู้ใช้ควรต้องชั่งน้ำหนักดูว่าประโยชน์ที่จะได้รับนั้นคุ้มกับเวลาและเงินที่จะต้องลงไปในการดัดแปลงโปรแกรมที่ใช้งานได้อยู่แล้วนี้หรือไม่

สำหรับผู้ที่เพิ่งจะเริ่มใช้ดีเบส ถ้าเลือกใช้ดีเบสทรีเลยก็จะสามารถเรียนได้เร็วขึ้น ทั้งนี้เพราะดีเบสทรีมีเมนู (The Assistant) ซึ่งผู้ใช้สามารถเลือกคำสั่งที่ต้องการได้ แทนที่จะต้องสั่งงานผ่าน Dot Prompt อย่างเดียวเช่นในดีเบสทู นอกจากนี้การที่เมนู Assistant แสดงตัวอย่างของโครงสร้างประโยคคำสั่ง ก็นับเป็นการช่วยให้ผู้ใช้เรียนรู้การใช้คำสั่งที่ถูกต้องได้เร็วขึ้น

ดีเบสทรีพัฒนาขึ้นมาเพื่อแก้ไขข้อจำกัดของดีเบสทู

สำหรับผู้ที่สามารถใช้ดีเบสทูได้ดีแล้ว แต่ในงานประจำวันยังคงต้องอาศัยการเขียนโปรแกรมด้วยแล้ว ก็ควรที่จะหันมาใช้ดีเบสทรี เพราะสามารถทำงานที่ระดับ Dot Prompt ได้มากกว่าและดีกว่าดีเบสทู เช่น ทำ report ซึ่งสามารถใช้ยอดรวมย่อย (subtotals) 2 ชั้น, ทำ Label, คิดคำนวณวันที่, เรียกใช้คำสั่ง Dos หรือโปรแกรมอื่นๆภายนอก ซึ่งถ้าเป็นดีเบสทูแล้วจะต้องอาศัยการเขียนโปรแกรมเข้าช่วย

การที่ดีเบสทรีเพิ่มคำสั่งใหม่ๆ เช่น Set Relation หรือการที่ดีเบสทรีพลัสมีคำสั่ง Catalog, View, Query, และ Field ช่วยให้การทำงานที่ต้องเปิดไฟล์หลายไฟล์พร้อมๆกันง่ายขึ้นมาก ทั้งนี้เพราะคำสั่ง Set Relation ช่วยให้สามารถเชื่อมไฟล์สองไฟล์ที่มีฟิลด์ใดฟิลด์หนึ่งเหมือนกันได้ โดยที่เมื่อมีการสร้างตัวชี้ (pointer) ไว้ที่เรคอร์ดใดเรคอร์ดหนึ่งแล้ว ดีเบสจะไปสร้างตัวชี้ตัวที่สองไว้ที่เรคอร์ดที่มีข้อมูลตรงกัน งานเช่นนี้ผู้ใช้ดีเบสทรีสามารถใช้คำสั่งที่ Dot Prompt ทำได้

เหตุผลสำคัญสองข้อที่ผู้ใช้จะเปลี่ยนจากดีเบสทู มาใช้ดีเบสทรี คือ ดีเบสทรีมีฟังก์ชันเฉพาะงานมากกว่า และประสิทธิภาพที่ถูกปรับปรุงขึ้นอย่างมาก

จะเห็นได้ว่ามีสาเหตุสำคัญ 2 ประการที่จูงใจให้ผู้ใช้งานส่วนใหญ่หันมาเลือกใช้ดีเบสทรีแทนดีเบสทู ประการแรกได้แก่การที่ดีเบสทูมีข้อจำกัดในการทำงานมากกว่าดีเบสทรี ประการที่สองได้แก่การที่แอชตัน–เทตได้ปรับปรุงประสิทธิภาพในการทำงานของดีเบสทรี จนเหนือกว่าดีเบสทูในเกือบทุกด้าน

ดีเบสทรีได้แก้ไขข้อจำกัดในการทำงานซึ่งผู้ใช้ดีเบสทูต้องประสบอยู่ เช่น จากการที่เรคอร์ดหนึ่งจะประกอบด้วยฟิลด์ได้ไม่เกิน 32 ฟิลด์และมีขนาดจุได้แค่ 1,000 ตัวอักษร มาเป็นเรคอร์ดขนาด 128 ฟิลด์และจุได้ 4,000 ตัวอักษร และจำนวนเรคอร์ดที่จะมีได้ในแต่ละไฟล์ได้ไม่เกิน 65,535 เรคอร์ดเป็นหนึ่งพันล้านเรคอร์ดในดีเบสทรี (แต่จะว่ากันตามจริงแล้ว การที่ดีเบสทูจำกัดจำนวนเรคอร์ดแค่ 65K ก็ไม่เป็นปัญหาอะไรมากนัก เพราะไมโครคอมพิวเตอร์และฮาร์ดดิสก์ทุกวันนี้ ก็ยังไม่สามารถจะทำงานได้ดีนักกับไฟล์ขนาดใหญ่ที่จุถึงพันล้านเรคอร์ด) และในส่วนของตัวแปรหน่วยความจำดีเบสทรีได้เพิ่มจำนวนจาก 64 เป็น 256 ตัวแปร และได้ขยายขนาดจาก 1,500 ตัวอักษร เป็น 31,744 ตัวอักษร และการที่ดีเบสทรีได้ปรับปรุงฟิลด์ชนิดตัวเลขคำนวณ ให้ใช้ตัวเลขได้ 16 หลักแทนที่จะเป็น 10 หลัก เช่นในดีเบสทู ช่วยให้การทำงานด้านการเงินและบัญชีสะดวกขึ้นมาก ดีเบสทรียังได้เพิ่มชนิดของฟิลด์อีก 2 ชนิดคือ ชนิดหน่วยความจำ (Memo Field) ซึ่งช่วยในการเก็บข้อความยาวๆ (Text) และชนิดวันที่ (Date Field) ซึ่งทำให้การเก็บข้อมูลตามลำดับเวลาเป็นได้อย่างสะดวก และการคิดคำนวณเกี่ยวกับจำนวนวันก็สามารถทำได้โดยไม่ต้องอาศัยการเขียนโปรแกรมช่วย

แต่ข้อจำกัดที่สำคัญที่สุดของดีเบสทูเห็นจะได้แก่การที่ผู้ใช้งานสามารถเปิดไฟล์ฐานข้อมูลพร้อมกันได้เพียง 2 ไฟล์ ในขณะที่ดีเบสทรีสามารถเปิดไฟล์ได้ 10 ไฟล์พร้อมกัน (แต่เป็นที่น่าสังเกตว่าดีเบสทรีมีการจำกัดการเปิดไฟล์อื่นๆ เช่น ไฟล์ดัชนี (index), ไฟล์รายงาน (Report) และไฟล์หน่วยความจำ (memo) ให้เปิดพร้อมกันได้เพียง 15 ไฟล์ แต่ในการทำงานจริงๆแล้ว การเปิดไฟล์ฐานข้อมูลพร้อมกันทีเดียว 10 ไฟล์แทบจะไม่ปรากฏ โดยปกติแล้วจะมีการเปิดไฟล์ฐานข้อมูลอย่างมากแค่ 4–5 ไฟล์ และไฟล์ดัชนีอีก 3–4 ไฟล์ ก็เพียงพอสำหรับการใช้งานจริงๆแล้ว)

เหตุผลประการที่สองที่จูงใจให้ผู้ใช้ส่วนใหญ่หันมาใช้ดีเบสทรี คือความ

ต้องการที่จะเพิ่มประสิทธิภาพในการทำงานของดีเบส แต่จะว่าไปแล้วก็เป็นเรื่องของความรู้สึกเฉพาะตัวบุคคลกันมากกว่า สำหรับผู้ที่คุ้นเคยกับการทำงานที่รวดเร็วของสเปรดชีท อาจจะรู้สึกว่าโปรแกรมภาษาดีเบสทำงานได้ช้าไม่ทันใจ ในขณะที่ผู้ที่ไม่เคยมีคอมพิวเตอร์ไว้ช่วยงานมาก่อนเลย อาจจะรู้สึกว่าโปรแกรมเดียวกันนี้ช่างวิเศษและช่วยให้งานเสร็จได้รวดเร็วเสียจริงๆ ตัวอย่างเช่น หากนำโปรแกรมดีเบสทูไปใช้ในการทำบัญชีรายรับและออกใบสั่งของ การออกใบสั่งแต่ละรายการอาจจะต้องหยุดรอเครื่องครั้งละ 10 วินาที เพื่อให้โปรแกรมเช็คเลขที่บัญชีลูกค้า รายการสินค้าจากบัญชีสินค้าคงคลัง และลงรายการนั้นต่อท้ายบัญชีใบสั่งของ การทำงานที่ต้องหยุดรอสัก 10 วินาที สำหรับพนักงานที่พิมพ์ดีดได้เร็วอาจจะรู้สึกว่านานเกินไป แต่สำหรับเสมียนที่ต้องทำทุกอย่างเองโดยไม่มีเครื่องคอมพิวเตอร์ช่วย เช่น ต้องเขียนใบสั่งใบส่งของ และลงบัญชีเอง อาจจะรู้สึกว่าโปรแกรมเดียวกันนี้ช่วยประหยัดเวลาและแรงงานได้มากทีเดียว

ผลการทดสอบความสามารถในการทำงานของดีเบสทั้ง 3 เวอร์ชัน

ได้มีการทดสอบความเร็วในการทำงานของดีเบสทั้ง 3 เวอร์ชัน โดยทำการทดสอบ 12 ชนิดซึ่งได้แสดงผลไว้ในตารางเปรียบเทียบ ปรากฏว่าดีเบสทรีสามารถทำงานได้เร็วกว่าดีเบสทูมาก ทั้งนี้เพราะดีเบสทรีสามารถนำหน่วยความจำ (RAM) ที่เกินจากที่โปรแกรมต้องการ (256K) มาใช้ จะเห็นได้ว่าดีเบสทรีทำเวลาได้ดีมากในการจัดลำดับข้อมูล (Sort) และสร้างดัชนี (Index) แต่ที่สำคัญที่สุดสำหรับผู้ที่เคยใช้ดีเบสทูก็คือ เมื่อมีการเพิ่มเรคอร์ด หรือเปลี่ยน Key Field ดีเบสทรีสามารถ update ไฟล์ดัชนีหลายๆ ไฟล์ได้เร็วกว่าดีเบสทูมาก

ดีเบสทรีพลัสมีคำสั่งที่ใช้ในการทำงานพร้อมกันหลาย ๆ งาน ซึ่งแต่ก่อนเป็นที่ต้องการของผู้ใช้อย่างมาก

ในโปรแกรมที่ใช้คำสั่ง @...SAY ...GET เพื่อเรียกใช้ฟิลด์และใช้คำสั่ง READ ดึงข้อมูล ดีเบสทรีสามารถทำงานได้ดีกว่าดีเบสทูอย่างเห็นได้ชัด ทั้งนี้เพราะในดีเบสทูเทคนิคในการค้นหา (Search) ข้อมูลในไฟล์ดัชนีอื่นๆ ยกเว้นไฟล์ดัชนีที่เปิดไฟล์แรกยังทำได้ไม่ดีพอ และถ้าเป็นการทำงานกับไฟล์ขนาดใหญ่ (10,000 เรคอร์ดหรือมากกว่าเป็นต้น) ผู้ใช้งานก็จะพบว่าแม้แต่การเปิดไฟล์ดัชนีมากกว่า 2 ไฟล์ในโปรแกรมที่จะต้องมีการเพิ่มเรคอร์ดใหม่ก็ไม่สะดวกเสียแล้ว

เปลี่ยนจากดีเบสทูมาเป็นดีเบสทรี-คุ้มหรือไม่

สิ่งแรกที่ผู้ใช้ดีเบสทูจะต้องลงทุนเมื่อหันมาใช้ดีเบสทรี ที่แน่นอนคือค่าซอฟท์แวร์ (สำหรับบ้านเราที่มีการก๊อบปี้กันอย่างแพร่หลายก็คงจะไม่หนักหนาเท่าไร ทั้งนี้ไม่นับรวมไปถึงการใช้งานแบบมัลติยูสเซอร์) แต่ผู้ที่ใช้ดีเบสทูกับระบบ CP/M หรือกับเครื่อง 8 บิท เห็นจะต้องลงทุนซื้อเครื่องขนาด 16 บิทอีก นอกจากนี้ยังต้องคำนึงถึงเวลาและค่าใช้จ่ายที่จะต้องให้พนักงานฝึกใช้เวอร์ชันใหม่ และค่าใช้จ่ายในการแปลงโปรแกรมเฉพาะงานให้ใช้ในดีเบสทรีได้นั้น หากจัดการได้ไม่ดีพอ ค่าใช้จ่ายในการนี้อาจจะแพงกว่าค่าเครื่องเสียอีก

การแปลงข้อมูล, ไฟล์, โปรแกรมในดีเบสทูเพื่อใช้ในดีเบสทรี

ในการแปลงไฟล์ระบบฐานข้อมูลจากดีเบสทูมาใช้งานในดีเบสทรีนั้น หากว่าการใช้งานเป็นไปแต่ในระดับการสั่งงานจาก Dot Prompt และระบบการใช้งานเป็นแบบซิงเกิ้ลยูสเซอร์ก็แทบจะไม่มีปัญหาอะไรเลย เพราะสามารถใช้ DCONVERT UTILITY ซึ่งสามารถแปลงไฟล์ฐานข้อมูล ไฟล์ฟอร์แมท ไฟล์สำหรับออกรายงานและไฟล์หน่วยความจำ หรือแม้แต่โปรแกรมเฉพาะงานของดีเบสทูมาเป็นดีเบสทรีได้ สำหรับไฟล์ดัชนีนั้นดีเบสจะสร้างโปรแกรมเบ็ดเสร็จซึ่งรวมคำสั่งที่จำเป็นในการสร้าง index ใหม่ให้โดยเป็นไปตาม Key expression อันเดิม DCONVERT UTILITY ใช้ง่ายและสะดวกดีเพราะเป็นเมนูสำเร็จให้เลือกสั่งตามต้องการได้

แต่การแปลงโปรแกรมเฉพาะงานซึ่งใช้ดีเบสทูเขียนขึ้นนั้นอาจจะทำแบบตรงไปตรงมาไม่ได้ เพราะคำสั่งและฟังก์ชันของดีเบสทรีต่างจากดีเบสทูอยู่มาก แต่พอจะถือได้ว่า DCONVERT ก็เป็นโปรแกรมที่ช่วยงานได้พอสมควรทีเดียว เพราะสามารถเลือกคำสั่งมาใช้แทนได้อย่างเหมาะสม เช่น จะเปลี่ยน $ ในดีเบสทูมาใช้ SUBSTR, @ เป็น AT, ! เป็น UPPER, # เป็น RECNO( ), * เป็น DELETED( ), ERASE เป็น CLEAR และ USING เป็น PICTURE

ดังนี้เป็นต้น และเมื่อ DCONVERT ไม่สามารถหาคำที่เหมาะสมมาแทนให้โดยอัตโนมัติแล้ว ก็จะช่วยแก้ปัญหาให้โดยการเปลี่ยนประโยคสั่งงานในโปรแกรมนั้นให้เป็น Comment Line แทน เพื่อว่าเมื่อนำโปรแกรมนั้นมาใช้ในดีเบสทรีแล้วจะไม่เกิด syntax error และผู้ใช้จะค้นหาประโยคนั้นเพื่อนำมาแก้ไขได้ง่ายขึ้น

แต่จะว่าไปแล้ว คงไม่มีโปรแกรมสำเร็จรูปใดจะสามารถทำการแปลงไฟล์หรือโปรแกรมเฉพาะงานได้ครบถ้วนสมบูรณ์ เพราะโครงสร้างและตรรกของโปรแกรมแต่ละโปรแกรมที่มีอยู่ย่อมถูกจำกัดโดยโครงสร้างของคำสั่งและการใช้งานที่แตกต่างกันของดีเบสทั้งสองเวอร์ชัน จะเห็นได้ว่าแม้ DCONVERT จะฉลาดพอที่จะเปลี่ยนคำสั่ง USING มาใช้ PICTURE แต่ก็จะไม่สามารถคาดการณ์ได้ว่า ถ้าเป็นการฟอร์แมทหรือคัดข้อมูลแล้ว หากเลือกใช้คำสั่ง RANGE หรือ @ Function แทน PICTURE แล้ว ผู้ใช้งานจะได้ประโยชน์มากขึ้น

ในโปรแกรมที่มีการเปิดใช้ไฟล์ฐานข้อมูล 2 ไฟล์ DCONVERT จะเปลี่ยนคำสั่ง SELECT PRIMARY และ SELECT SECONDARY เป็น SELECT A และ SELECT B และแปลงข้อมูลที่เกี่ยวข้องของฟิลด์ต่างๆ ในส่วนที่ไม่ได้ใช้งานจาก P และ S ให้เป็น A-> และ B-> ตามลำดับ อย่างไรก็ตามผู้ใช้งานอาจจะชอบที่จะใช้ตัวเลขในการกำหนดส่วนงานที่โปรแกรมดึงมาใช้มากกว่าที่จะใช้ A และ B เพราะเป็น

ตารางเปรียบเทียบความสามารถในการทำงานของดีเบส

ชนิดของแบบทดสอบ	จำนวนครั้งของงานที่ทำ	ดีเบสทู	ดีเบสทรี	ดีเบสทรีพลัส
1. นับ 1-1000	1000	35	63	59
2. นับ 1-1000 และแสดงตัวเลขแต่ละตัว	1000	99	129	114
3. อ่านค่าและประเมินข้อความ (Complicated Character String)	1000	80	197	193
4. ใช้คำสั่ง @..SAY ให้แสดงผลทางจอภาพ	100	235	296	258
5. ใช้คำสั่ง TEXT..ENDTEXT แสดงผลทางจอภาพ	100	448	203	227
6. ใช้คำสั่ง @..GET เรียกฟิลด์จำนวน 21 ฟิลด์	100	74	139	141
7. อ่านไฟล์ขนาดจุ 1000 เรคอร์ดตามลำดับข้อมูลที่ป้อนเข้าไป (Sequential)	1	66	64	42
8. อ่านข้อมูลขนาด 1000 เรคอร์ดตามลำดับดัชนี	1	133	121	93
9. เพิ่ม (Add) เรคอร์ด 1 เรคอร์ด โดยเปิดไฟล์ดัชนีไว้ 1 ไฟล์	100	102	135	129
10. เพิ่ม (Add) เรคอร์ด 1 เรคอร์ด โดยเปิดไฟล์ดัชนีไว้ 7 ไฟล์	100	1419	1048	941
11. สร้างไฟล์ดัชนีขึ้นใหม่ 7 ไฟล์	1	1708	1299	291
12. จัดลำดับ (Sort) 1200 เรคอร์ด โดยใช้ Key ขนาด 10 ตัวอักษรเป็นตัวกำหนด	1	1944	371	134

หมายเหตุ การทดสอบนี้ใช้ทดลองกับดีเบสทูเวอร์ชัน 2.4 ดีเบสทรีเวอร์ชัน 1.1 และดีเบสทรีพลัสเวอร์ชัน 1.0 โดยใช้เครื่องไมโครคอมพิวเตอร์ Compaq Portable ซึ่งใช้ Hard Disk แบบแยกส่วนยี่ห้อ Bernoulli เวลาที่ทำการทดสอบบันทึกไว้เป็นวินาที ฐานข้อมูลที่ใช้คือ Mailing List ซึ่งเดิมมีอยู่ 1,000 เรคอร์ด (ได้เพิ่มเป็น 1,200 เรคอร์ดก่อนที่จะทำการทดสอบในข้อ 9 และข้อ 10) แต่ละเรคอร์ดมีความยาว 301 ตัวอักษร ในแบบทดสอบข้อ 9 และ 10 ได้ใช้คำสั่ง APPEND BLANK และ REPLACE เพื่อเพิ่มจำนวนเรคอร์ดให้เป็น 1,200 เรคอร์ด

การสะดวกกว่าที่จะตั้งชื่อแฝง (Alias Name) ของไฟล์ให้ใกล้เคียงชื่อเดิม เพื่อว่าจะได้ไม่ต้องวุ่นอยู่กับการจดจำว่าไฟล์ใดใช้งานอยู่ส่วนไหนของโปรแกรม ดังนี้เป็นต้น

ถึงแม้ว่า DCONVERT UTILITY จะเป็นประโยชน์และช่วยในการแปลงไฟล์และโปรแกรมในดีเบสทูให้ใช้งานในดีเบสทรีได้ดี แต่ก็มีหลายๆ อย่างที่ DCONVERT ทำไม่ได้ เช่น ในดีเบสทูมีการใช้คำสั่ง FIND เมื่อแปลงเป็นดีเบสทรีแล้ว ก็ยังคงใช้ FIND ตามเดิม ซึ่งถ้าเปลี่ยนเป็น SEEK แล้วจะได้ประโยชน์มากกว่า ดังนี้เป็นต้น ดังนั้นผู้ใช้งานเองก็จะต้องใช้เวลามากพอสมควร ในการดัดแปลงทดสอบและแก้ไขข้อบกพร่องของโปรแกรมที่ได้ดัดแปลงมา ก่อนที่จะได้นำไปใช้งานจริงๆ สำหรับโปรแกรมเฉพาะงานซึ่งใช้ได้ดีในดีเบสทูแล้ว ก็อาจจะเป็นการดีกว่าที่จะใช้งานเช่นนี้ต่อไปตามเดิม หรือเรียกว่า "ถ้ายังดีๆ อยู่ ก็อย่าไปยุ่งให้มากเรื่องหรือเรื่องมากเลย"

แต่ก็อาจจะมีในหลายกรณีที่การแปลงไฟล์และโปรแกรมดีเบสทูมาเป็นดีเบสทรีเป็นเรื่องที่เลี่ยงไม่ได้ อาจจะเป็นนโยบายที่บริษัทฯ กำหนดมาให้ว่าจะต้องเลิกใช้ระบบ CP/M เพราะไม่ต้องการที่จะใช้ระบบการทำงาน 2 ระบบ หากเป็นเช่นนี้แล้ว ทางออกที่ดีที่สุดที่จะทำได้รวดเร็วที่สุดและประหยัดที่สุดเห็นจะได้แก่การแปลงไฟล์ที่มีอยู่ให้เป็นไฟล์ดีเบสทรีด้วย DCONVERT และแก้ไขโปรแกรมที่ใช้อยู่ให้พอใช้งานได้

อย่างไรก็ตาม หากผู้ใช้มีความประสงค์ที่จะปรับปรุงประสิทธิภาพการทำงานของโปรแกรมซึ่งโครงสร้างเดิมมีข้อจำกัดอยู่มาก เช่นในเรื่องของจำนวนฟิลด์มีได้ไม่เกิน 32 ฟิลด์ หรือ 1,000 ตัวอักษร หรือเปิดไฟล์พร้อมกันได้เพียง 2 ไฟล์นั้น อาจจะทำได้โดยการเปลี่ยนโครงสร้างของฐานข้อมูลใหม่ ซึ่งหมายถึง

หากต้องการใช้งานในระบบมัลติยูสเซอร์และต้องการการรักษาความปลอดภัยของข้อมูลก็ต้องเลือกใช้ดีเบสทรีพลัส

การที่จะต้องดัดแปลงโปรแกรมย่อยใหม่เสียเป็นส่วนใหญ่ เช่นในกรณีที่แต่เดิมผู้เขียนโปรแกรมต้องสร้างฟิลด์ Address ให้เป็นฟิลด์ยาวฟิลด์เดียวแทนที่จะซอยแบ่งเป็น 3 ฟิลด์ ทั้งนี้เพราะดีเบสทูจำกัดจำนวนฟิลด์แค่ 32 ฟิลด์ และต้องใช้ SUBSTR ฟังก์ชันเรียกข้อมูลส่วนที่ต้องการมาใช้ และเนื่องจากการเปิดไฟล์ฐานข้อมูลก็จำกัดเพียง 2 ไฟล์ ผู้เขียนโปรแกรมก็อาจจะต้องนำข้อมูลซึ่งน่าจะได้แยกไว้คนละไฟล์มารวมไว้ด้วยกัน เช่น นำเอารายการสั่งซื้อ และรายการเรียกเก็บเงินมารวมไว้ในไฟล์เดียวกันเป็นต้น

เนื่องจากดีเบสทูมีข้อจำกัดหลายประการด้วยกัน ไฟล์ที่สร้างขึ้นจึงมีข้อมูลซึ่งซ้ำซ้อนอยู่มาก เช่น ในระบบบัญชีรายรับหากต้องการที่จะพิมพ์รายการ Invoice จะต้องมีการสร้างไฟล์ Invoice ซึ่งจะมีรายชื่อลูกค้า และที่อยู่ของลูกค้าไว้ด้วย เพราะเมื่อนำมาออก Report ก็จะได้เปิดไฟล์เพียง 2 ไฟล์คือ ไฟล์ Invoice และไฟล์ Transaction ซึ่งถ้าเป็นดีเบสทรีแล้วก็จะสามารถประหยัดเวลาและเนื้อที่ใช้งาน Disk ได้เพียงแค่เพิ่มหมายเลขบัญชีลูกค้าไว้ในไฟล์ Invoice และให้เรียกชื่อและที่อยู่ของลูกค้าจากไฟล์ที่สามหรือไฟล์บัญชีรายชื่อลูกค้านั่นเอง ในการทำงานบางอย่างซึ่งดีเบสทรีทำได้อย่างง่ายๆ ด้วยการเปิดไฟล์พร้อมกัน 3 ไฟล์ (เช่น การลงบัญชีในไฟล์ลูกค้าและไฟล์บัญชีแยกประเภท) ซึ่งถ้าใช้ดีเบสทูแล้ว การที่จะทำงานเช่นว่านี้ได้จะต้องเปิดไฟล์ทีละ 2 ไฟล์ ซึ่งเสียเวลามาก หรือไม่ก็เปิดไฟล์หนึ่งยืนไว้ เรียกอีกไฟล์หนึ่งมาใช้และปิดไป แล้วเรียกไฟล์ที่ 3 มาทำงานต่อซึ่งทำให้เสียเวลามากขึ้นไปอีก

ในกรณีที่มีฟิลด์ซึ่งสร้างขึ้นเพื่อเก็บข้อมูลที่มีขนาดยาวมากๆ ซึ่งนานๆ จะได้ใช้สักครั้งนั้น ถ้าเป็นในดีเบสทรีแล้ว ผู้ใช้งานสามารถประหยัดเวลาได้มากกว่าโดยสร้างฟิลด์ดังกล่าวนี้ให้เป็นฟิลด์ชนิดหน่วยความจำ (Memo) และการที่ดีเบสทรีได้เพิ่มฟิลด์ชนิดวันที่อีกด้วย ก็ช่วยให้การคิดคำนวณที่เกี่ยวกับวันที่ทำได้โดยไม่ยุ่งยากและไม่ต้องอาศัยการเขียนโปรแกรมเข้าช่วยเช่นในดีเบสทู ซึ่งในการคำนวณหาจำนวนวันระหว่างวันที่สองวันที่กำหนดมานั้น ดีเบสทูจะต้องสร้างอัลกอริทึมที่ซับซ้อนมากพอดู

ข้อที่น่าสังเกตคือ การแปลงโครงสร้างของฐานข้อมูลที่มีอยู่เพื่อให้ใช้ได้กับโครงสร้างที่ปรับปรุงใหม่นั้น แม้ว่าจะสามารถทำได้โดยใช้ DCONVERT และคำสั่งอื่นๆ ประกอบด้วย เช่น COPY, MODIFY STRUCTURE, REPLACE อาจจะกินเวลาและแรงงานมากพอๆ กับการที่มาเริ่มตั้งต้นสร้างไฟล์กันใหม่เลยทีเดียว

ถ้าหากว่ามีโปรแกรมเฉพาะงานซึ่งจำเป็นต้องมีการเขียนบางส่วนใหม่ก็อาจจะถือว่ามีส่วนดีอยู่เหมือนกันเพราะ

การที่ต้องมาแก้ไข หรือรื้อโครงสร้างใหม่นั้น ช่วยให้ผู้ใช้เข้าใจโครงสร้างของโปรแกรมได้ลึกซึ้งและขณะเดียวกันก็ได้เรียนรู้สิ่งใหม่ๆที่ดีเบสทรีเพิ่มเติมมาอีกด้วย

ในส่วนของการเขียนโปรแกรมนั้น ดีเบสทูไม่เอื้ออำนวยให้โปรแกรมเมอร์ทำงานได้สะดวกเท่าใดนัก จะเห็นได้ว่าโดยปกติแล้วโปรแกรมเมอร์จะสร้างโครงสร้างเป็นโมดูลย่อยๆก่อน เพราะสะดวกและรวดเร็วในการเรียกใช้ แต่ในดีเบสทูลักษณะเช่นนี้กลับทำให้ดีเบสทูทำงานได้ช้าลงมากกว่าการสร้างเป็นไฟล์ขนาดใหญ่เพียง 2-3 ไฟล์ เพราะในแต่ละครั้งที่ดีเบสทูดึงโปรแกรมย่อยมาใช้ ก็จะต้อง Load ตัวโปรแกรมจาก disk ไปไว้ใน memory ทุกครั้งไป อันที่จริงแล้ว โปรแกรมเมอร์สามารถสร้างโมดูลย่อยหลายๆอันแล้วนำมาผูกเป็นโปรแกรมใหญ่อีกทีหนึ่งได้ แต่จะยุ่งยากและเสียเวลามากหากต้องมีการเปลี่ยนแปลงโปรแกรมนั้นในภายหลัง จึงไม่คุ้มที่จะใช้วิธีนี้สร้างโปรแกรม

ดีเบสทรีแก้ปัญหานี้โดยการเพิ่มไฟล์ Procedure ซึ่งสามารถเก็บโปรแกรมย่อยได้ถึง 32 ชุดและสามารถเรียกมาใช้งานได้รวดเร็ว นอกจากนั้นโปรแกรมเมอร์ยังสามารถผ่านข้อมูลไปยังโปรแกรมอื่นหรืองานส่วนอื่นได้โดยอาศัย parameter ซึ่งทำให้สามารถสร้างโมดูลอเนกประสงค์ซึ่งดีเบสสามารถเรียกใช้ได้จากโปรแกรมต่างชนิดกัน และหากว่าผู้ใช้งานมีโครงการที่จะใช้งานลักษณะเดียวกันซ้ำหลายหนแล้ว หากลงทุนลงแรงสร้าง Procedure Files เก็บเป็น Library ไว้ ก็จะช่วยทุ่นเวลาได้มาก

จากดีเบสทรีมาเป็นดีเบสทรีพลัส

การเปลี่ยนจากดีเบสทรีมาใช้ดีเบสทรีพลัสนั้นยุ่งยากน้อยกว่าการเปลี่ยนจากดีเบสทูมาใช้ดีเบสทรีหรือดีเบสทรีพลัส

เพราะไฟล์ทุกชนิดในดีเบสทรีสามารถนำไปใช้งานในดีเบสทรีพลัสได้เลยทันที ทั้งสองเวอร์ชันจะต่างกันก็ตรงที่ดีเบสทรีพลัสได้เพิ่มคำสั่ง ฟังก์ชัน และสิ่งใหม่หลายอย่าง ซึ่งหมายความว่าการเปลี่ยนจากดีเบสทรีมาใช้ดีเบสทรีพลัสเป็นขบวนการที่สะดวกและสบายทีเดียว จะว่าไปแล้วดีเบสทรีพลัสมีคำสั่งและฟังก์ชันทุกอย่างที่ดีเบสทรีมี ต่างกันก็ตรงที่ดีเบสทรีพลัสมีมากกว่านั่นเอง

นอกจากการปรับปรุงรูปโฉมให้น่าดูน่าใช้ขึ้นแล้ว คู่มือดีเบสทรีพลัสยังเขียนได้ดีกว่า ละเอียดมากกว่า ดีเบสทรีพลัสมีการพูดถึงข้อจำกัดในการทำงาน และแนะนำเกร็ดต่าง ๆ ที่เป็นประโยชน์ ซึ่งถ้าเป็นก่อนหน้านี้ผู้ใช้งานจะต้องเสียเวลาลองใช้ผิด ๆ ถูก ๆ อยู่หลายชั่วโมงกว่าจะจับจุดได้ถูก

ดีเบสทรีพลัส = ดีเบสทรี + คำสั่งและฟังก์ชันใหม่ ๆ

ดีเบสทรีพลัสยังส่งเสริมการใช้คำสั่งที่ Dot Prompt ให้เป็นไปอย่างง่าย ๆ ซึ่งหากเป็นเมื่อก่อน งานเช่นเดียวกันนี้จะต้องอาศัยการเขียนโปรแกรมเข้าช่วย การที่ดีเบสทรีพลัสมีคำสั่งใหม่ ๆ เพิ่มเข้ามาเช่น CATALOG, VIEW, QUERY, และสามารถกำหนดสภาพงาน (Environment) และจำนวนและชนิดของไฟล์ที่ต้องการนำมาใช้งาน

คำสั่ง Catalog จะแสดงรายการของไฟล์ที่จัดอยู่ในประเภทเดียวกันซึ่งย่อมรวมถึงไฟล์ฐานข้อมูล และไฟล์อื่น ๆ ที่เกี่ยวกับไฟล์ฐานข้อมูล (ไฟล์ดัชนี, ไฟล์ฟอร์แมท, ไฟล์ query และแบบสำหรับออกรายงานและ Label เป็นต้น)

การเลือกใช้คำสั่ง View File จาก Assist Menu หรือจาก Dot Prompt ช่วยในการเรียกกลุ่มไฟล์ที่มีส่วนสัมพันธ์กันมาใช้งานพร้อม ๆ กัน ซึ่งจะเป็นการเรียกใช้ไฟล์ฐานข้อมูลเพียงไฟล์เดียวหรือหลายไฟล์ก็ได้ บวกกับไฟล์ดัชนีอีกจำนวนหนึ่ง และสามารถกำหนดเนื้อที่ทำงานและฟิลด์ที่จะเรียกมาใช้งาน หรืออาจจะเรียกใช้ไฟล์ฟอร์แมทและทำการสร้าง condition ได้ตามต้องการ

นอกจากนี้คำสั่ง Set Field To ยังสามารถเรียกกลุ่มฟิลด์จากฐานข้อมูลทั้งหมดที่กำลังเปิดใช้งานอยู่มาใช้งานหรือตรวจแก้ได้ ซึ่งคำสั่งนี้นอกจากจะช่วยให้สามารถระบุขอบเขตของฟิลด์จากหลาย ๆ ไฟล์มาใช้งานได้สะดวกโดยที่ไม่ต้องใช้คำสั่งซ้ำ ๆ กันทุกครั้ง ยังช่วยในการดึงฟิลด์จากส่วนงาน (Work Area) อื่นมาใช้และแก้ไขได้เพียงใช้คำสั่ง EDIT, BROWSE, @..SAY...GET และ REPLACE

ดีเบสทรีพลัสยังมีคำสั่งใหม่ ๆ ที่ทำให้การสั่งงานที่ Dot Prompt ง่ายและสะดวกขึ้นมาก ทั้งนี้รวมถึงการกำหนดขนาดของ Buffer ที่เก็บข้อความขนาด 32,000 ตัวอักษรไว้ล่วงหน้า นอกจากนั้นดีเบสทรีพลัสยังนำเอาวิธีการของดีเบสทูที่โต้ตอบกับผู้ใช้งานเมื่อผู้ใช้งานใช้คำสั่งผิด ๆ (ซึ่งในดีเบสทรีได้เลิกใช้ไป) กลับมาใช้อีก แต่ก็ได้มีการปรับปรุงให้ดีขึ้นมาก ในดีเบสทรีพลัสยังสามารถที่จะเก็บประวัติย้อนหลังคำสั่งที่ใช้บ่อยครั้งที่สุด 20 คำสั่ง (อันที่จริงแล้วสามารถเพิ่มกำหนดให้เก็บประวัติถอยหลังไปได้ถึง 16,000 ครั้ง) และผู้ใช้งานสามารถแก้ไขเปลี่ยนแปลงคำสั่งเหล่านี้ได้

ดีเบสทรีพลัสได้เพิ่มฟังก์ชันใหม่ ๆ เช่น ในทางคณิตศาสตร์มีฟังก์ชัน ABS ( ) ใช้ในการหาค่า Absolute, MAX ( ) และ MIN ( ) ใช้ในการหาค่ามากที่สุดและน้อยที่สุด และ MOD ใช้หาผลหารของเลขสองจำนวน, ฟังก์ชัน IIF ( ) ใช้ในการหาค่าเชิงตรรกและแสดงผลจากการหาค่านั้น ๆ ทางจอภาพหรือเครื่องพิมพ์ได้ตามต้องการ, ฟังก์ชัน TRANSFORM ช่วยในการแปลงข้อมูลซึ่งหากเป็นเมื่อก่อนจะทำได้ก็ต้องอาศัยคำสั่ง @...SAY ...PICTURE เท่านั้น

นอกจากนี้ดีเบสทรีพลัสได้เพิ่มฟังก์ชันใหม่ ๆ ที่เป็นประโยชน์สำหรับการเขียนโปรแกรมเฉพาะงาน เช่นเมื่อมีการใช้คำสั่ง FIND หรือ SEEK แล้ว จะใช้ IF FOUND ( ) แทน IF. NOT. EOF ( ). ส่วนคำสั่ง ( ) ERROR และ ( ) MESSAGE ช่วยให้โปรแกรมที่เขียนขึ้นใช้ง่ายและไม่พังง่าย ๆ เพราะฟังก์ชันนี้จะชี้ให้ทราบว่าได้เกิด ERROR ขึ้นที่จุดใดและจะแจ้งรายละเอียดของปัญหาให้ทราบคร่าว ๆ ส่วน ON ERROR, ON ESCAPE, ON KEY เป็นคำสั่งที่ช่วยให้โปรแกรมทำงานได้รวดเร็วและตอบโต้กับผู้ใช้งานได้อย่างฉับไว

การใช้ข้อมูลร่วมกัน (Multi-user) และการป้องกันข้อมูล แอชตัน-เทตเสนอให้คุณในดีเบสทรีพลัส

มีอยู่สองกรณีที่ผู้ใช้ดีเบสจำเป็นจะต้องเปลี่ยนมาใช้ดีเบสทรีพลัสโดยไม่มีทางเลือก กรณีแรกคือ **เมื่อมีผู้ใช้งานหลายคนต้องใช้โปรแกรมหรือข้อมูลร่วมกัน** ซึ่งหมายถึงความจำเป็นที่จะต้องติดตั้งระบบเครือข่ายไมโครคอมพิวเตอร์ (Local Area Network) และกรณีที่สองคือ **เมื่อมีความจำเป็นที่จะต้องสร้างระบบความปลอดภัยของข้อมูลหรือระบบป้องกันข้อมูล** เพื่อไม่ให้ผู้อื่นมีส่วนรู้เห็นหรือนำไปใช้ได้ ดังนั้นบริษัทขนาดเล็กซึ่งมีพนักงานไม่กี่คนอาจจะไม่จำเป็นต้องใช้ระบบป้องกันข้อมูล แต่เลือกใช้ระบบเครือข่ายไมโครคอมพิวเตอร์ (LAN) เพื่อให้พนักงานได้ใช้ข้อมูลร่วมกัน ในขณะที่บริษัทขนาดใหญ่อาจจะไม่มีการใช้ข้อมูลร่วมกัน แต่มีความจำเป็นที่จะต้องมีระบบความปลอดภัยของข้อมูล ไม่ว่าจะเป็นกรณีหนึ่งกรณีใดข้างต้นนี้ ผู้ใช้งานจำเป็นจะต้องติดตั้ง dBase Administrator ซึ่งเป็นหัวใจของระบบเครือข่ายของดีเบสทรีพลัส

ดีเบสทรีพลัสสร้างระบบความปลอดภัยไว้ 3 ขั้นตอนด้วยกัน ผู้ใช้จะเข้าสู่ระบบงานได้ก็ต่อเมื่อสามารถระบุชื่อที่ถูกต้องของกลุ่มชื่อของฐานข้อมูล ชื่อผู้ใช้งานเอง และรหัสผ่าน (Password) ผู้ควบคุมการใช้งานจะต้องเป็นผู้กำหนดกลุ่มชื่อของฐานข้อมูล ชื่อไฟล์ที่ผู้ใช้จะเรียกขึ้นมา และจะต้องมีการกำหนดด้วยว่าผู้ใช้งานแต่ละคนสามารถใช้ส่วนใดของระบบ กล่าวคือ ไฟล์และฟิลด์ใดบ้างที่ผู้ใช้งานคนนั้นสามารถจะเพิ่มเติม ลบทิ้ง อ่าน หรือแก้ไขส่วนใดได้บ้าง เช่นนี้เป็นต้น และหากต้องการจะเพิ่มการรักษาความปลอดภัยให้กับไฟล์ฐานข้อมูลและไฟล์ดัชนีก็สามารถทำได้โดยกำหนดรหัสเฉพาะไฟล์นั้น ๆ ได้

เมื่อได้ติดตั้ง dBase Administrator แล้ว ผู้ใช้งานหลายคนก็สามารถที่จะใช้ซอฟท์แวร์และโปรแกรมใช้งานจากเครื่องของตนพร้อม ๆ กันได้ แต่หากเป็นการเรียกใช้งานที่ Dot Prompt ผู้ใช้งานควรจะระวังถึงปัญหาที่จะเกิดขึ้นได้เมื่อผู้ใช้งาน 2 คนเรียกเรคอร์ดเดียวกันขึ้นมาแก้ไขและ update พร้อม ๆ กันแล้ว ข้อมูลของผู้ที่ทำการ update ทีหลังเท่านั้นที่ได้ถูกบันทึกลงไฟล์ไว้ การป้องกันปัญหาเช่นนี้ทำได้โดยการล็อคไฟล์ทั้งไฟล์หรือเฉพาะเรคอร์ดนั้น ไม่ให้คนอื่น ๆ เรียกใช้ได้จนกว่าการ update ข้อมูลในส่วนนั้น ๆ จะเสร็จสิ้นลง

จะว่าไปแล้วดีเบสทรีพลัสก็แก้ไขปัญหาได้โดยอัตโนมัติอยู่แล้ว โดยจะล็อคไฟล์ทั้งไฟล์ไว้เมื่อมีการเรียกใช้คำสั่ง APPEND, INDEX, REPLACE ALL และ UPDATE ส่วนการใช้คำสั่ง EDIT หรือ CHANGE เพื่อเปลี่ยนแปลงแก้ไขเรคอร์ด ผู้ใช้ควรจะล็อคเรคอร์ดนั้นไว้ก่อนโดยใช้คำสั่ง RLOCK หรือ กด Control-O ซึ่งสามารถสั่งที่ Dot Prompt ได้ทันที

ในโปรแกรมใช้งานของดีเบสทรีพลัส การล็อคไฟล์หรือล็อคเรคอร์ดอาจจะเป็นเรื่องที่ผู้ใช้งานมองไม่เห็น ฟังก์ชัน FLOCK ( ) ช่วยล็อคไฟล์ทั้งไฟล์เพื่อให้สร้างงานเป็นชุด (Batch type) ได้ ในขณะที่คำสั่ง RLOCK ( ) จะใช้ในการล็อคเรคอร์ดใดเรคอร์ดหนึ่งเพื่อทำการแก้ไข แต่การที่จะเพิ่มขีดความสามารถในการทำงานแบบ Multi-User เข้ามาในระบบการทำงานของดีเบสที่มีอยู่แต่เดิมแล้ว ย่อมหมายถึงการแก้ไขโปรแกรมใหม่เพื่อว่าเมื่อมีการ update ข้อมูลโดยใช้คำสั่ง Replace หรือเมื่อมีการใช้คำสั่ง @... SAY...GET และตามด้วยคำสั่ง READ จะมีการล็อคเรคอร์ดนั้นไว้ก่อนที่จะทำการ update และยกเลิกคำสั่งล็อคนั้นเมื่อเสร็จงานที่ต้องการแล้วเพื่อที่ผู้ใช้งานอื่นจะได้เรียกใช้บ้าง

เมื่อท่านผู้อ่านได้อ่านมาถึงตรงนี้ก็คงจะตัดสินใจได้แล้วว่า จะเลือกใช้ดีเบสเวอร์ชันไหนจึงจะตอบสนองความต้องการของท่านได้ดีที่สุด อย่างไรก็ตาม แอชตัน-เทต ก็คงจะพัฒนาดีเบสต่อไปเพื่อให้รับกับเทคโนโลยีทางฮาร์ดแวร์และความต้องการของผู้ใช้ เมื่อถึงตอนนั้น (อาจจะเป็นดีเบสโฟร์หรืออะไรก็แล้วแต่) เรา ๆ ท่าน ๆ ก็คงจะต้องเลือกกันใหม่อีก และเกิดบทความนี้ขึ้นมาอีก (ภาคสอง) พบกันใหม่โอกาสหน้าค่ะ